Analysis of QIIME2 output

Lev Litichevskiy

January 12, 2023

The goal of this notebook is to make several key plots based on the output of QIIME2, specifically the counts matrix and the taxonomy annotations. QIIME2 has plug-ins for doing all of these analyses, but I prefer to do them in R.

The plots that we will make are:

1) PCoA
2) Barplots at different taxonomic levels
3) Alpha diversity
4) TODO: identifying taxons different between two groups with ANCOM or DESeq2

Search the document for "CHANGEME" to see which lines of code will require project-specific tweaks.

This demo dataset contains gut microbiome samples from young, middle-aged, and old mice before and after undergoing fecal microbial transfer (FMT) of a young microbiome. Before FMT, we expect to see samples separate by age, but after FMT, the samples look quite similar to each other because they received the same FMT input.

Load libraries

```
library(tidyverse)
library(cowplot)
library(ggpubr)
library(phyloseq)
library(speedyseq) # to speed up the tax_glom function
library(vegan)

# use black-and-white theme, and increase default font size
theme_set(theme_bw(base_size=15))
```

Import metadata

This file should contain any important metadata about your samples of interest.

```
(meta.df <- read.table("data/demo_qiime2_metadata.tsv", sep="\t", header=T))
```

```
##    Sample.ID    Age Timepoint
## 1         S1 Middle   Pre-FMT
## 2         S2 Middle  Post-FMT
## 3         S3 Middle   Pre-FMT
## 4         S4 Middle  Post-FMT
## 5         S5 Middle   Pre-FMT
## 6          S6 Middle  Post-FMT
## 7          S7 Middle   Pre-FMT
## 8          S8 Middle  Post-FMT
## 9          S9    Old   Pre-FMT
## 10        S10    Old  Post-FMT
## 11        S11    Old   Pre-FMT
## 12        S12    Old  Post-FMT
## 13        S13    Old   Pre-FMT
## 14        S14    Old   Pre-FMT
## 15        S15    Old  Post-FMT
## 16        S16  Young   Pre-FMT
## 17        S17  Young  Post-FMT
## 18        S18  Young   Pre-FMT
## 19        S19  Young  Post-FMT
## 20        S20  Young   Pre-FMT
## 21        S21  Young  Post-FMT
## 22        S22  Young   Pre-FMT
## 23        S23  Young   Pre-FMT
## 24        S24  Young  Post-FMT
## 25        S25  Young   Pre-FMT
```

- The sample identifier should be in a column called `Sample.ID`
- If it's not, you can search and replace `Sample.ID` with whatever else (e.g. `sample_id`)

Import taxonomy

```
tax.df <- read.table("data/demo_qiime2_taxonomy.tsv", sep="\t", header=T)

# create new column for each taxonomy level
tax.df <- tax.df %>%
  separate(Taxon, c("kingdom", "phylum", "class", "order", "family", "genus", "species"),
           sep="; [[:alpha:]]__", remove=T, fill="right")

# take a peek at the taxonomy
tax.df[1:3, ]
```

```
##                         Feature.ID     kingdom       phylum       class
## 1 9c9bfd420fffa8772f8982255054c692 d__Bacteria Bacteroidota Bacteroidia
## 2 1cb85e10fa890fda99460d5703c18abd d__Bacteria   Firmicutes     Bacilli
## 3 abcda143e8418f4e2b05862424b1a1a2 d__Bacteria Bacteroidota Bacteroidia
##             order            family          genus                     species
## 1   Bacteroidales   Bacteroidaceae    Bacteroides Bacteroides_thetaiotaomicron
## 2 Lactobacillales Lactobacillaceae  Lactobacillus      Lactobacillus_johnsonii
## 3   Bacteroidales   Muribaculaceae Muribaculaceae          uncultured_bacterium
##   Confidence
## 1  0.9987711
## 2  0.9943885
## 3  0.7073147
```

Import counts

In the default output of QIIME2, the second line of the counts table starts with `#` before OTU.ID, which causes R to treat this line as a comment. You will need to delete the `#` at the start of the second line before trying to import into R.

```
counts.df <- read.table("data/demo_qiime2_feature_table.tsv", sep="\t", header=T)

# set the taxon names as the rownames
counts.df <- column_to_rownames(counts.df, var="OTU.ID")

# take a peek at the counts
counts.df[1:5, 1:5]
```

```
##                                    S1    S2   S3    S4    S5
## 9c9bfd420fffa8772f8982255054c692  114  6687  118  7437    50
## 1cb85e10fa890fda99460d5703c18abd 4994 43562 5507 84937 10130
## abcda143e8418f4e2b05862424b1a1a2 4998 10860 5338 42894  2206
## 1bb73fe7e3fdfcc431d59cc403af855f  402    17  870    34  2576
## a2213eede305d5a387ec2a9e56412657    9  6330   11   112     0
```

```
dim(counts.df)
```

```
## [1] 2593   25
```

We have 2593 taxons x 25 samples.

Confirm that all samples are in the metadata

```
all(colnames(counts.df) %in% meta.df$Sample.ID)
```

```
## [1] TRUE
```

(optional) Filtration

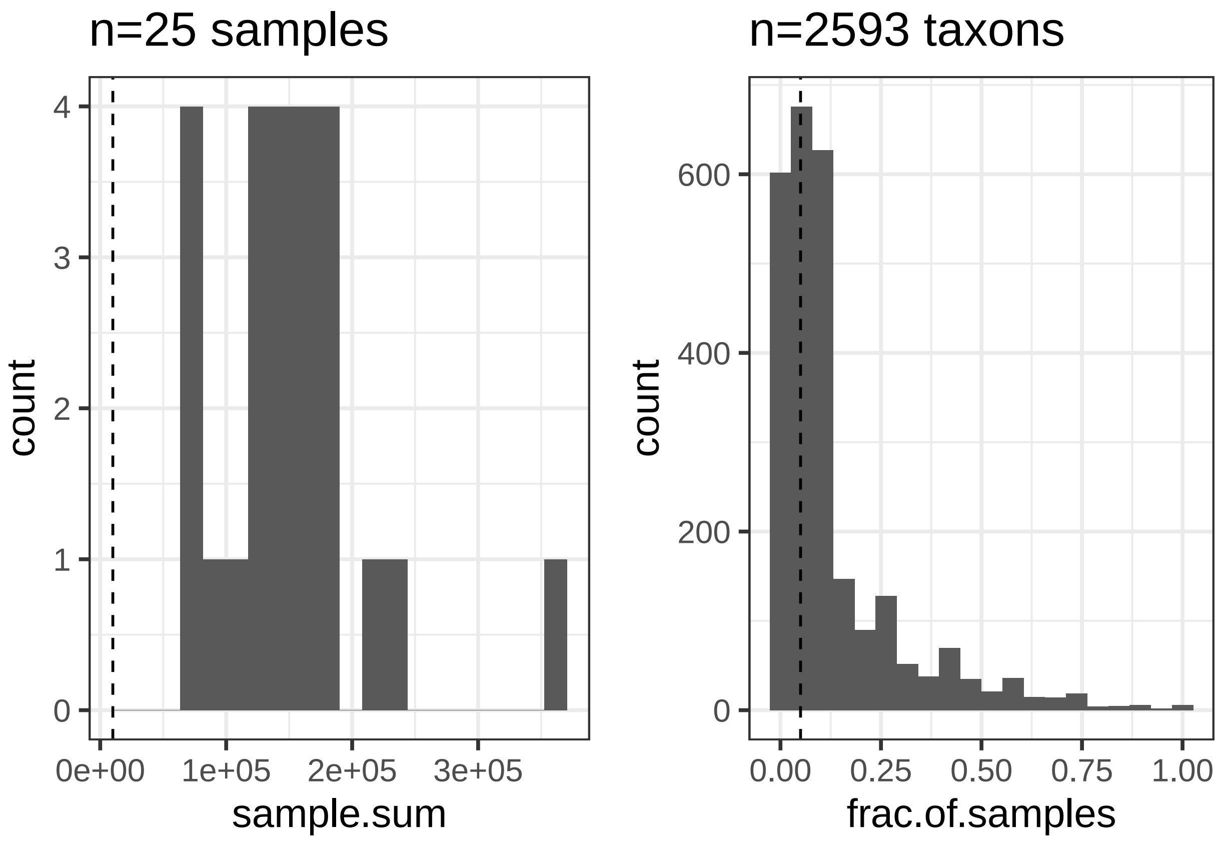
Sometimes, we need to discard samples that received too few reads. Or we want to discard taxons that we detected very rarely because we think they're noise or contamination. This section can help you do that.

For example, say that we want to keep samples with at least 10k total counts and taxons detected in at least 5% of samples. These thresholds are arbitrary: change them as necessary.

Visualize filtration thresholds

```
# CHANGEME if you want to change filtration thresholds
SAMPLE.SUM.THRESHOLD <- 10000
TAXON.FRAC.THRESHOLD <- 0.05

p1 <- data.frame(sample.sum=colSums(counts.df)) %>%
  ggplot(aes(sample.sum)) +
  geom_histogram(bins=20) +
  geom_vline(xintercept=SAMPLE.SUM.THRESHOLD, lty=2) +
  labs(title=sprintf("n=%i samples", ncol(counts.df)))

p2 <- data.frame(frac.of.samples=rowSums(counts.df > 0)/ncol(counts.df)) %>%
  ggplot(aes(frac.of.samples)) +
  geom_histogram(bins=20) +
  geom_vline(xintercept=TAXON.FRAC.THRESHOLD, lty=2) +
  labs(title=sprintf("n=%i taxons", nrow(counts.df)))
```

```
plot_grid(p1, p2, nrow=1)
```



Based on these thresholds, we'll keep all samples and discard lots of taxa.

Do filtering

```
samples.to.keep <- colnames(counts.df)[colSums(counts.df) > SAMPLE.SUM.THRESHOLD]
taxons.to.keep <- rownames(counts.df)[rowSums(counts.df > 0)/ncol(counts.df) > TAXON.FRAC.THRESHOLD]
counts.filt.df <- counts.df[taxons.to.keep, samples.to.keep]
```

```
dim(counts.df)
```

```
## [1] 2593   25
```

```
dim(counts.filt.df)
```

```
## [1] 1315   25
```

With these thresholds, we go from 2593 to 1315 taxons, and we don't discard any samples.

I will use the **unfiltered** counts dataframe for the rest of this notebook. If you want to use the filtered data, you'll need to tweak the metadata and taxonomy dataframes (see next code chunk).

Create phyloseq object

We combine all our data into a phyloseq object because the phyloseq package has a number of useful functions that we want to use.

```r
# if using the filtered dataframe
# tmp.physeq.meta.df <- meta.df %>% column_to_rownames("Sample.ID")
# physeq.meta.df <- tmp.physeq.meta.df[colnames(counts.filt.df), ]
# tmp.physeq.tax.df <- tax.df %>% column_to_rownames("Feature.ID")
# physeq.tax.df <- tmp.physeq.tax.df[rownames(counts.filt.df), ]

# if using the unfiltered dataframe
physeq.meta.df <- meta.df %>% column_to_rownames("Sample.ID")
physeq.tax.df <- tax.df %>% column_to_rownames("Feature.ID")

physeq <- phyloseq(
  counts.df %>% as.matrix() %>% otu_table(taxa_are_rows=T), # can substitute counts.filt.df here
  physeq.meta.df %>% sample_data(),
  physeq.tax.df %>% as.matrix() %>% tax_table()
)

physeq
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 2593 taxa and 25 samples ]:
## sample_data() Sample Data:       [ 25 samples by 2 sample variables ]:
## tax_table()   Taxonomy Table:    [ 2593 taxa by 8 taxonomic ranks ]:
## taxa are rows
```

Aggregate to different taxonomy levels

```r
physeq.phylum <- physeq %>% tax_glom(taxrank="phylum")
physeq.class <- physeq %>% tax_glom(taxrank="class")
physeq.family <- physeq %>% tax_glom(taxrank="family")
physeq.genus <- physeq %>% tax_glom(taxrank="genus")
physeq.species <- physeq %>% tax_glom(taxrank="species")
```

```r
ntaxa(physeq.phylum)
```

```
## [1] 10
```

```r
ntaxa(physeq.class)
```

```
## [1] 15
```

```r
ntaxa(physeq.family)
```

```
## [1] 70
```

```r
ntaxa(physeq.genus)
```

```
## [1] 146
```

```r
ntaxa(physeq.species)
```

```
## [1] 261
```

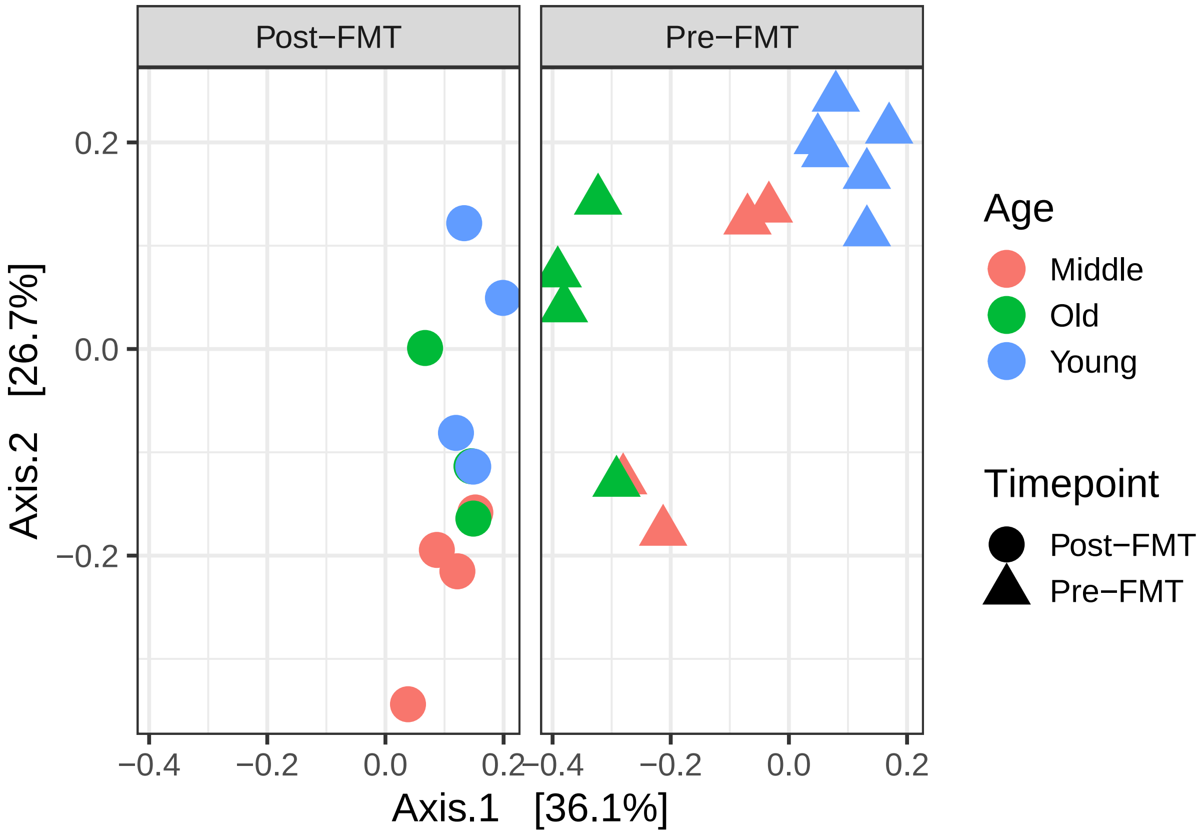
I like to use genus-level data.

PCoA

```
pcoa.genus.res <- physeq.genus %>%

  # compute relative abundance
  transform_sample_counts(function(OTU) OTU/sum(OTU)) %>%

  # perform PCoA using Bray-Curtis distance
  ordinate(method="MDS", distance="bray")
```

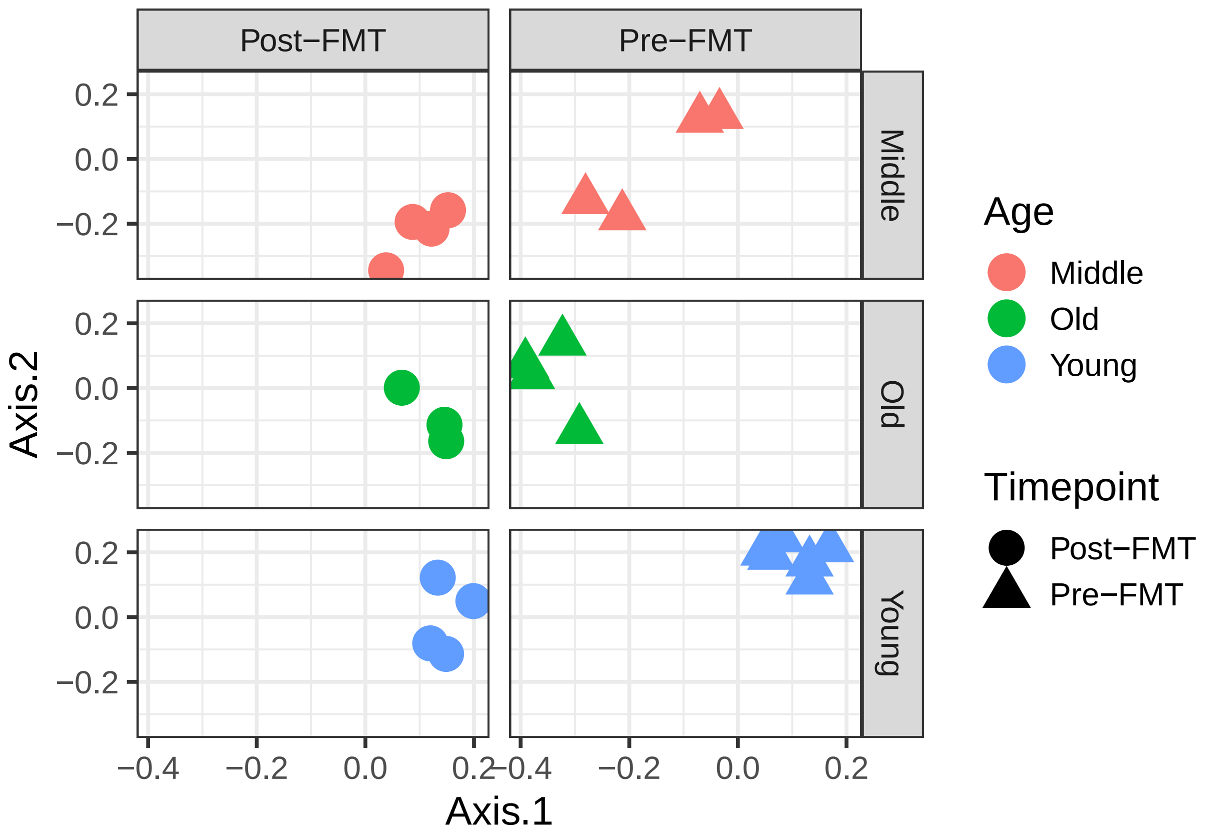
We can use the `plot_ordination` function to make a plot.

```
# CHANGEME: adjust color and shape as desired
plot_ordination(physeq.genus, pcoa.genus.res, type="samples", color="Age", shape="Timepoint") +
  geom_point(size=6) +
  facet_wrap(~Timepoint)
```



We see good separation by age before FMT, but not as much separation after FMT. We can also directly get the PCoA coordinates and make a custom plot ourselves.

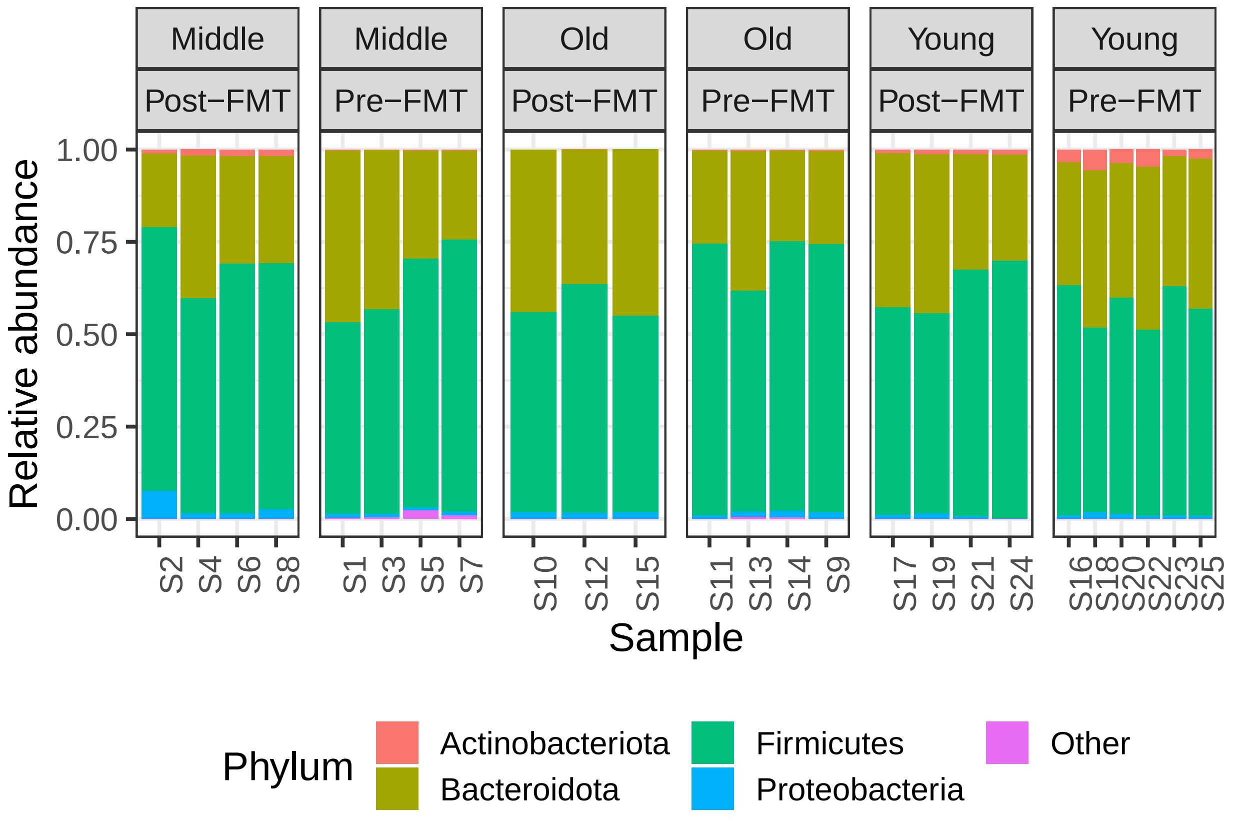
```
# get PCoA coordinates
pcoa.genus.df <- plot_ordination(physeq.genus, pcoa.genus.res, type="samples", justDF=T)

# CHANGEME: change color, shape, and facet to your variables of interest
pcoa.genus.df %>%
  ggplot(aes(x=Axis.1, y=Axis.2, color=Age, shape=Timepoint)) +
  geom_point(size=6) +
  facet_grid(Age~Timepoint)
```



Barplots

Phylum

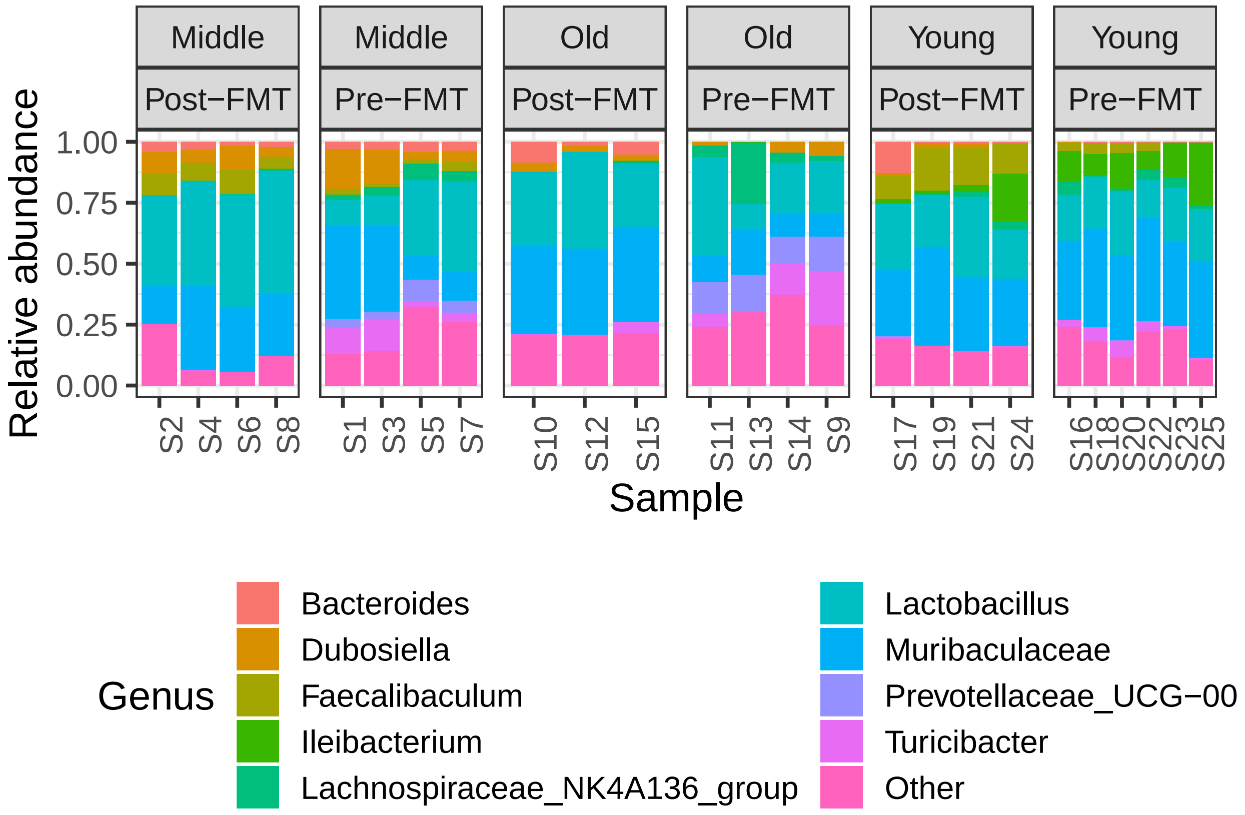
```
# identify the top 4 most abundant phyla
# this is especially important for lower taxonomic levels that
# have too many groups to show in one plot
top.n.phyla <- data.frame(tax_table(physeq.phylum))[
  names(sort(taxa_sums(physeq.phylum), decreasing=T))[1:4], "phylum"]

physeq.phylum %>%

  # convert to df
  psmelt %>%

  # aggregate less common phyla into "Other"
  mutate(agg.phylum=fct_relevel(
    ifelse(phylum %in% top.n.phyla, phylum, "Other"), "Other", after=Inf)) %>%

  # compute sum of counts per sample
  group_by(Sample) %>%
  mutate(total.abund=sum(Abundance)) %>%

  # compute sum of counts per sample-phylum group
  group_by(Sample, agg.phylum, total.abund) %>%
  summarise(abund=sum(Abundance), .groups="drop") %>%
  # compute relative abundance
  mutate(rel.abund=abund/total.abund) %>%

  # add back metadata
  merge(data.frame(sample_data(physeq.phylum)), by.x="Sample", by.y="row.names") %>%

  # plot
  ggplot(aes(x=Sample, y=rel.abund, fill=agg.phylum)) +
  geom_bar(stat="identity") +
  labs(y="Relative abundance", fill="Phylum") +

  # put legend below plots and rotate x-labels
  theme(legend.position="bottom",
        axis.text.x=element_text(angle=90, hjust=1)) +
  guides(fill=guide_legend(nrow=2)) +

  # CHANGEME: modify as necessary
  facet_wrap(~Age+Timepoint, nrow=1, scales="free_x")
```



Genus

```
# identify the 9 most abundant genera
top.n.genera <- data.frame(tax_table(physeq.genus))[
  names(sort(taxa_sums(physeq.genus), decreasing=T))[1:9], "genus"]

physeq.genus %>%
```

```r
# convert to df
psmelt %>%

# aggregate less common genera into "Other"
mutate(agg.genus=fct_relevel(
  ifelse(genus %in% top.n.genera, genus, "Other"), "Other", after=Inf)) %>%

# compute sum of counts per sample
group_by(Sample) %>%
mutate(total.abund=sum(Abundance)) %>%

# compute sum of counts per sample-genus group
group_by(Sample, agg.genus, total.abund) %>%
summarise(abund=sum(Abundance), .groups="drop") %>%

# compute relative abundance
mutate(rel.abund=abund/total.abund) %>%

# add back metadata
merge(data.frame(sample_data(physeq.genus)), by.x="Sample", by.y="row.names") %>%

# plot
ggplot(aes(x=Sample, y=rel.abund, fill=agg.genus)) +
geom_bar(stat="identity") +
labs(y="Relative abundance", fill="Genus") +

# put legend below plots and rotate x-labels
theme(legend.position="bottom",
      axis.text.x=element_text(angle=90, hjust=1)) +
guides(fill=guide_legend(nrow=5)) +

# CHANGEME: modify as necessary
facet_wrap(~Age+Timepoint, nrow=1, scales="free_x")
```



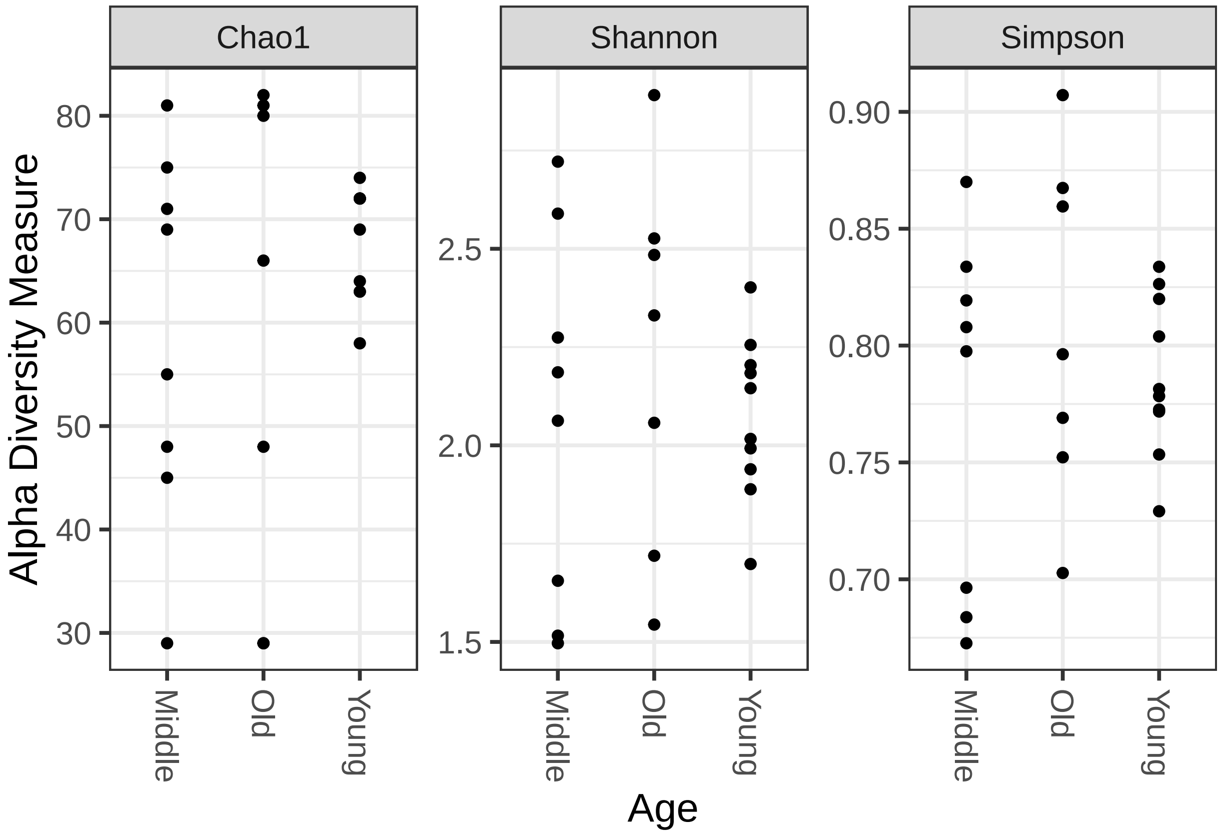
Alpha diversity

Genus

We can use the `plot_richness` command from `phyloseq`.

```
# CHANGEME: change x to be your variable of interest
plot_richness(physeq.genus, x="Age", measures=c("Chao1", "Shannon", "Simpson"))
```

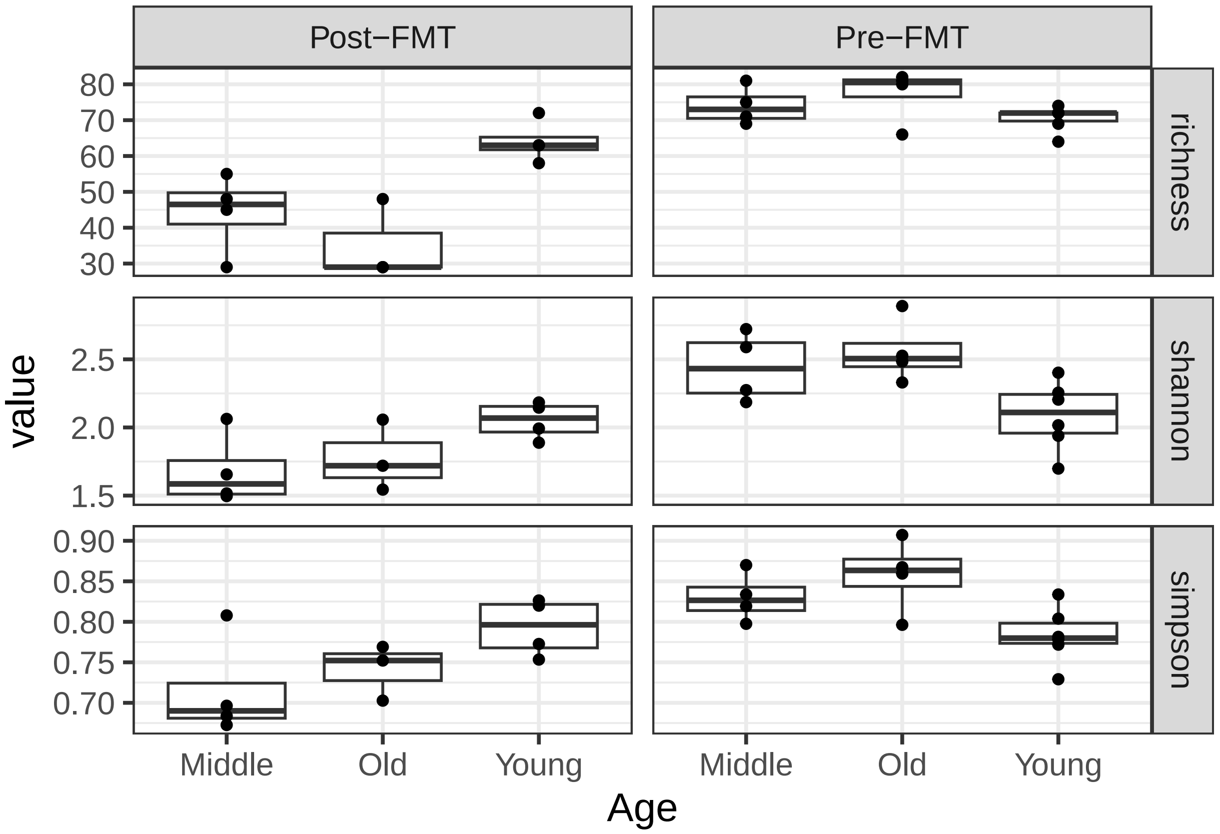
```
## Warning in estimate_richness(physeq, split = TRUE, measures = measures): The data you have provided d
## any singletons. This is highly suspicious. Results of richness
## estimates (for example) are probably unreliable, or wrong, if you have already
## trimmed low-abundance taxa from the data.
##
## We recommended that you find the un-trimmed data and retry.
```



Or compute the diversity metrics manually.

```
alpha.div.genus.df <- data.frame(
  shannon=diversity(otu_table(physeq.genus), index="shannon", MARGIN=2),
  simpson=diversity(otu_table(physeq.genus), index="simpson", MARGIN=2),
  richness=colSums(otu_table(physeq.genus) != 0)) %>%
  rownames_to_column("Sample.ID") %>%
  pivot_longer(c(shannon, simpson, richness), names_to="diversity", values_to="value") %>%
  merge(meta.df, by="Sample.ID")
```

```
# CHANGEME: adjust plot as desired
alpha.div.genus.df %>%
  ggplot(aes(x=Age, y=value)) +
  geom_boxplot(outlier.shape=NA) +
  geom_point() +
  facet_grid(diversity ~ Timepoint, scales="free_y")
```



TODO: Use ANCOM or DESeq2 to identify taxons different between two groups

I will get to this some day. The phyloseq website has a decent tutorial about this: https://joey711.github.io/phyloseq-extensions/DESeq2.html.